

#### Wojciech Mioduszewski

# Klasyfikacja danych opisanych za pomocą szeregów czasowych

Master's Thesis

Supervisor: dr inż. Jerzy Błaszczyński

Poznań, 2015

## Contents

1	Wst	бb																	3
2	Bacl	kground																	5
	2.1	Section																	5
		2.1.1 Su	ıbsecti	ion															5
	2.2	Inline for	mattin	ng															5
	2.3	Special cl	naracte	ers															5
	2.4	Figures																	6
	2.5	Tables .																	6
	2.6	Source co	de exa	ampl	les.														7
	2.7	Math .																	8
	2.8	Algorithm	ns																9
	2.9	Bibliogra	phy .	•											•				9
3	Con	cept and	Desig	gn o	f tł	<b>ie</b>	Sy	rst	en	1									11
4	Imp	lementat	ion																<b>13</b>
5	Perf	ormance	Evalu	ıati	on														<b>15</b>
6	Con	clusions																	17
$\mathbf{A}$	Usei	rs Guide																	19
Bil	oliogr	anhv																	21

$\sim$ 1		- 1
( )	napter	
$\sim$ 1	iapici	_

## Wstęp

The Introduction may be put before the  $\mbox{\mbox{\tt mainmatter}}$  command which will disable numbering of this chapter while still adding to the table of contents.

The goal and the scope of the thesis

## Background

The thesis can be structured using the following sectioning styles:

#### 2.1 Section

#### 2.1.1 Subsection

#### 2.1.1.1 Subsubsection

**Paragraph** 

Subparagraph

#### 2.2 Inline formatting

We suggest using *Insets*, like:

**strong** for strong emphasizing some text.

*emph* for emphasizing some text.

**Code** for formatting of names of modules, procedures, class names, variables, etc.

path for formatting of file names and directories, like /usr/share/doc/packages/

texlive-latex. The names are properly broken at the ends of lines. However, path names containing special LATEX characters must be typeset using ERT and

the \dcspath command, e.g. sample\_file.

**kbd** for formatting of shortcuts, e.g.: **Ctrl-c**.

cmd for formatting system commands.

name for formatting other special names.

#### 2.3 Special characters

1. Non-breaking space can be inserted using Ctrl-space. It produces "~" in LATEX code.

6 2 Background

2. A normal, inter-word space can be inserted using Ctrl-Alt-space. It produces "\" in LATEX code. This type of space is useful for formatting spacing after dots, e.g. here. By default LATEX produces here a longer space used for separating whole sentences.

- 3. A thin space can be produced by Ctrl-Shift-space, e.g. here. It produces "\," in LATEX code.
- 4. Sentence-ending space can be inserted using Ctrl-., which produces "\@." in LATEX code. This type of space is useful in sentences ending with a capital letter. In such cases LATEX recognizes the last word as a acronym and places a regular inter-word space instead of inter-sentence space. Consider the following example:

This can be achieved by using HTTP. This protocol...

5. Hyphenation indicator can be inserted using Ctrl- –, which is used for marking possible places of hyphenation, e.g. democracy.

### 2.4 Figures

The figures should be put in floats, like Fig. 2.1. You can also reference figures using prettyref package like this: Fig. 2.1.



Figure 2.1: Example figure

It is possible to combine several pictures inside one float. Just insert a float inside a float. See Fig. 2.2 for example. Please note the horizontal spacing between subfigures.

#### 2.5 Tables

Tables should have captions above like Table 2.1. Use small sans-serif fonts inside tables.

2.6 Source code examples 7

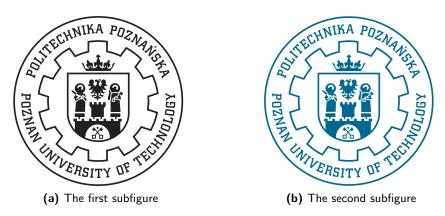


Figure 2.2: Example figure

Table 2.1: Example table

Column 1	Column 2	Column 3
One	1	4
Two	2	5
Three	3	6

### 2.6 Source code examples

There are a few different methods of including sample codes:

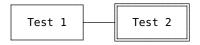
1. Using standard LyX-Code style:

```
#include <stdio.h>
int main() {
  printf("Hello world!\n");
  return 0;
}
```

Note 1: Empty lines must contain at least one single space to remain visible.

Note 2: There is no way to activate automatic syntax highlighting inside LyX-Code. However, you can use normal inline formatting inside.

Note 3: Lyx-Code can contain special characters, so it can be used to produce some ASCII art, e.g.:



2. By inserting *Program Listing*:

```
#include <stdio.h>
int main() {
  printf("Hello world!\n");
```

8 2 Background

```
return 0;
}
```

Note: By default the lstlisting environment does not add any left margin. You can change it by adding xleftmargin in the Settings > Advanced dialog box, e.g.:

```
procedure sayHello()
```

3. By inserting LATEX Code (ERT block) and using  ${\tt codeblock}$  environment:

```
#include <stdio.h>
int main() {
  printf("Hello world!\n");
  return 0;
}
```

- 4. The listings package can produce floats by itself. See Listing. 2.1 for example.
- 5. And finally, You can include code from external file:

```
\documentclass[11pt,a4paper,polish,thesis]{dcsbook}
\usepackage[utf8]{inputenc}
\usepackage{babel}
\setcounter{secnumdepth}{4}
\setcounter{tocdepth}{3}
\begin{document}
```

#### 2.7 Math

Can be put in line like this:  $S = \sum_{i=1}^{i=K} x_i^2$  or in dedicated lines:

$$S = \sum_{i=1}^{i=K} x_i^2$$

Listing 2.1: The Hello World program in C

```
#include <stdio.h>
int main() {
  printf("Hello world!\n");
  return 0;
}
```

2.8 Algorithms 9

The equations can be also numbered like equation 2.1.

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2}$$
 (2.1)

## 2.8 Algorithms

Use dcsalg package or directly algorithmicx package.

### 2.9 Bibliography

The bibliography can be included in the thesis like in this case. You can then cite the publications like this [1]. The other (more professional) solution is to use  $BibT_EX$ . See LyX User's Guide for details.

Chapter 3	

# Concept and Design of the System

Chapter 4

# Implementation

Chapter 5	

## Performance Evaluation

Ch	apter 6		

## Conclusions

Appendix A	
F F	

## Users Guide

# Bibliography

[1] A. Tanenbaum. Operating Systems Design and Implementation. Prentice Hall, 2006.